

REMARKS/ARGUMENTS

Upon entry of this amendment, which amends claims 1, 11, and 21, and adds new claims 31-40, claims 1-40 will be pending. In the Office Action, the Examiner rejected claims 1-30 under 35 USC § 103(a) as being unpatentable over Erimli, et al. (U.S. Patent No. 6,618,390 B1, hereinafter "Erimli") in view of McKenney (U.S. Patent Application No. 2002/0194436 A1). Applicants respectfully request reconsideration of the claims in view of the amendments above and remarks below.

Claim 1 was rejected under 35 USC § 103(a) as being unpatentable Erimli and McKenney. The rejection states that Erimli discloses every element of claim 1 except for the element of "temporarily restoring at least one free cell pointer into internal cache configured to assist in the lowering a frequency of reads from and writes to the pointer RAM". McKenney is cited as disclosing this element.

Erimli discloses a network switch that includes a first random access memory (RAM) and a second RAM. The second RAM includes buffers that are used to store data frames. The first RAM includes indications of whether free buffers are available for use. If free buffers in the second RAM are available for use, buffer pointers are then created only when necessary to address buffers for storing data frames. *See, Erimli*, col. 2, lines 51-67, col. 3, lines 1-7. The buffer pointers are generated and then placed in a queue. *See, Erimli*, col. 12, lines 53-55. The buffer pointers may then be used to store data frames in buffers associated with the pointers.

McKenney discloses a system for providing a software approach to the weak memory consistency model. A pointer to a linked list is used to determine data that should be written from a read CPU to a write CPU.

Erimli and McKenney, either alone or in combination, fail to disclose or suggest "storing free cell pointers into a pointer random access memory (RAM), wherein each free cell pointer points to a memory buffer that is vacant and available for storing data traffic." Erimli discloses searching the first RAM to determine an indication of whether free buffers are available for use. If free buffers are available, the indication is used to generate a buffer pointer.

Accordingly, Erimli does not disclose or suggest storing free cell pointers into a pointer random access memory. Rather, an indication of whether free buffers are available for use is stored in the first RAM.

The rejection states that storing free cell pointers into a pointer RAM is taught as storing buffer pointers into a free buffer queue 64 in Erimli. Applicants submit that this does not disclose or suggest storing free cell pointers into a pointer random access memory.

Also, even if storing free cell pointers in the free buffer queue 64 is considered storing free cell pointers into a pointer RAM (it is not), then the element of "sending one or more free cell pointers to one or more queues of the external integrated circuit, the one or more free cell pointers including the at least one free cell pointer temporarily stored in the internal cache, wherein each free cell pointer in the one or more free cell pointers in the one or more queues is configured to become a write cell pointer" is not disclosed or suggested. If the buffer queue 64 is considered a pointer RAM in Erimli, then Erimli does not disclose or suggest the element of sending the free cell pointers to the queue because they are already stored in the queue. Additionally, nowhere in Erimli is it disclosed or suggested that the buffer pointers sent to the queue include "at least one free cell pointer temporarily stored in the internal cache".

It also would not have been obvious to store free cell pointers in the first RAM in Erimli. In fact, Erimli teaches against storing free cell pointers in the first RAM. The main purpose of Erimli is to not store buffer pointers in RAM. Erimli solves this problem by storing only an indication of whether free buffers are available for use in the first RAM. The buffer pointers are then generated only when necessary to address buffers for storing data frames. The generated buffer pointers are then sent to the on chip queue. Accordingly, Erimli does not disclose or suggest storing free cell pointers into a pointer random access memory.

Additionally, Erimli and McKenney, Erimli and McKenney, either alone or in combination, fail to disclose or suggest:

temporarily storing at least one free cell pointer from the pointer RAM
into internal cache configured to assist in lowering a frequency of reads from and
writes to the pointer RAM;

...

sending one or more free cell pointers to one or more queues of the external integrated circuit, the one or more free cell pointers including the at least one free cell pointer temporarily stored in the internal cache, wherein each free cell pointer in the one or more free cell pointers in the one or more queues is configured to become a write cell pointer.

The rejection states that McKenney discloses temporary storing at least one free cell pointer into internal cache as the pointer stored in the odd or even numbered cache line. Applicants submit that McKenney does not disclose or suggest temporarily storing at least one free cell pointer in internal cache. Rather, McKenney discloses a pointer to a data structure. McKenney does not disclose or suggest that the pointer is temporarily stored in internal cache. Rather, the pointer is used to point to a new element to be read in a data structure. *See McKenney*, page 3, par. 30. Nowhere in McKenney is it disclosed or suggested that the pointer is stored in a cache. Further, nowhere in the McKenney is it disclosed or suggested that the pointer is from stored free cell pointers being stored in a pointer random access memory. In fact, nowhere in McKenney is a pointer RAM disclosed or suggested.

Additionally, it would not have been obvious to combine an Erimli and McKenney. First, there is no suggestion in Erimli to include a cache to temporarily store at least one free cell pointer from a pointer RAM. Rather, Erimli is directed towards storing an indication of whether free buffers are available for use and then buffer pointers are generated and sent to a queue. The purpose of Erimli is to generate buffer pointers when necessary before they are sent to the queue, thus reducing the space required to maintain free buffer information. *See Erimli*, col. 3, lines 2-4. Accordingly, Erimli teaches against adding a cache to store buffer pointers and also teaches against temporarily storing buffer pointers in the cache.

Accordingly, Applicants respectfully request withdrawal of the rejections of claim 1. Claims 2-11 depends on the claim 1 and thus derive patentability at least therefrom.

Claims 11 and 21 should be allowable for at least a similar rationale as recited with respect to claim 1. Claims 12-20 depend from claim 11 and claims 22-30 depend from claim 21 and thus derive patentability at least therefrom.

Appl. No. 10/045,288
Amdt. dated September 8, 2004
Reply to Office Action of March 11, 2004

PATENT

New Claims 31-40

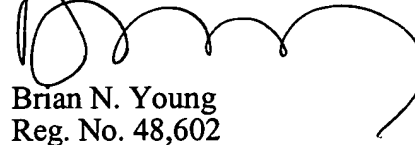
Applicant submits that the cited references do not disclose or suggest every element of new claims 31-40.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 415-576-0200.

Respectfully submitted,



Brian N. Young
Reg. No. 48,602

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 415-576-0200
Fax: 415-576-0300
BNY:jtc
60249544 v1